Correlated Data Listing
Page application
Tue, Feb 1, 2005

Demonstrating the ability to capture correlated data across front ends as means of assisting detailed 15 Hz studies, a new page application is being written. This note develops the details of that application and describes its operation.

Correlated data is easy for front ends to achieve. This page application CORR is designed to output a listing for each 15 Hz cycle for which selected data is acquired. Parameters that are input via the page display include a clock event for which the data is collected and a list of device names, from which readings are obtained for listing. The clock event can be 0x0C, which occurs for all 15 Hz cycles, or it can be 0x10, which occurs for all beam cycles, or it can be an individual Booster reset clock event, such as 0x1D, which is used for MiniBooNE beam cycles. The listing will show the current cycle Booster reset event as well as the readings of the selected analog channels in engineering units.

Use keyboard interrupts to enter the analog names, or the node and channel numbers, for each of up to 13 devices. If a name is entered, it is translated to a node:chan via name lookup. If a node:chan is entered, the corresponding 6-character name is shown alongside. As a page application, it keeps the node and channel values in its PAGEM memory, which only uses 48 bytes of the 120 bytes available. Interrupt on the row used for the specified clock event to start the data collection and listing. On that same line is the listing node#. Here is an example of the display layout:

```
          0         1         2         3
          01234567890123456789012345678901

          X CORRELATED DATA 12/21/04 1331
          *EV<10>  1D  N=    135 LIST<0509>
          GR2MID 0612:0102       0.988 NORM
          RF01GE 06B1:0117       48.94 KV
          (etc)
```

Besides the clock event for sampling the data, the reset event of the most recent set of data, a count of the number of lines of data collected, and the node# for targeting the serial listing is shown. The character immediately following "<10>" may be used to indicate the status code returned from GetDat. On the device lines is shown the readings of the selected devices, even though they are also output to the listing.

If one changes the device on a line, the collection of data stops. Likewise, an interrupt on the second line while the listing is active will stop the listing. That the listing is active is shown by inverse video on the "*" character. To start/stop the listing, interrupt on any of the first 6 character positions of the second line.

The listing format is as follows:

```
          0         1         2         3               9
          0123456789012345678901234567890123456789 ... 01234567

          EVENT   GR2MID RF01GE
           0x14    0.985  47.55
           -x1D    0.988  48.94
```

The initial minus sign in the last row indicates that the 0x1D cycle immediately follows the 0x14 cycle, both of which are beam events, so they are included in the event 0x10 spec. A line contained readings for all 13 devices uses 7 + 13*7 = 98 characters, plus the CR, LF.

Manage a list of devices for which data is sampled. One can enter a name or enter a `node:chan`, depending upon the location of the cursor. Whichever is chosen, the other is determined by a one-shot request for which the code can simply await the reply. If an entered name is blank, delete that device from list and move those below up one line, blanking the last line. If an interrupt occurs in the first column of a line, a blank line is inserted, and those that follow are moved down one line, eliminating the one that was on the last line, if any, and blanking what was on the line before. When activation occurs, the list of `node:chan` values is compressed so that there are no blank lines. The count of devices is thus determined for the subsequent event-based data request.

The data request is made using listype 40, which obtains engineering units values for all devices. When a line is initialized, it always includes the engineering units text, which is obtained whenever a `node:chan` pair is set.

Use `PAGECLAS` as a starting point for developing this PA.

Measure the elapsed time for each 15 Hz execution for which `HaveEvt` returns true, beginning after the return from `GetDat`. When the `N=` field is updated, check a mode flag for showing instead the `E=` value, which is this elapsed time. If it is in μs, `CvI` can be used for both cases. Toggling of this internal flag is done by interrupt under the 2-char field.

There will be an array of channel names and an array of `node:chan` pairs. The latter is kept in `PAGEM` space, so it is remembered across callups of the page. During PA initialization, this array is collected, and names (and engineering units text) are obtained via a one-shot request.

As for keyboard interrupts, the `*` field causes activate/deactivate. In the `EV<xx>` field of the clock event number, an interrupt causes deactivation, if necessary, reading the event#, and activation using that event. During any activation, read the listing node# field, which is also retained in `PAGEM` space. The meaning of an interrupt on the device lines depends on the column position. If it is in the name area, but not the first column, read the name and do a name lookup to get the `node:chan`. If it is in the `node:chan` field, read it and lookup the channel name (and engineering units). All of these one-shot requests can simply await the reply, a luxury that is afforded a PA.

Plan to allow a field to show the status return code, as a status of 7 means that the data may not be correlated. It can also happen if the node running the PA is not synchronized with the 15 Hz operation of the target nodes.

Usage summary

Assemble a set of names or `node:chan` pairs (maximum of 13) as the devices for which correlated data is to be gathered and listed. Insert a blank line by clicking on the first column. Blank a line by typing a blank in the first column and clicking. Select a clock event number of the 15 Hz cycles for which the data is to be sampled. Choose a target node# to which the serial port data is sent. Interrupt (click) on the early part of the event number line to start the collection. The `*` at the beginning of that line lights up to show ongoing activity. Click on that `*` when ready to terminate the activity. Readings of the devices are shown on the display as the values are collected for output to the listing. The Booster reset clock event is displayed and also presented in the first field of each line of the listing. An indication of cycles that are consecutive is shown by a — sign ahead of the event number. A line showing data from a cycle that is not consecutive with the previous line is indicated with a `0` in that same position.